

CLAIMS

What is claimed is:

1. 1. A method for constructing a software application from a model representation, the method comprising steps of:
 - 3 - reading the model representation;
 - 4 - identifying in the model representation a plurality of software modules;
 - 5 - identifying a programming language for each one of the plurality of software modules;
 - 6 - compiling each one of the plurality of software modules into machine language using a software compiler, the software compiler corresponding to the identified programming language; and
 - 7 - linking the compiled plurality of software modules into the software application.
8. 2. The method of claim 1 wherein the step of reading the model representation further comprises the step of reading the model representation from one or more electronic files.
9. 3. The method of claim 1 wherein the step of reading the model representation further comprises step of identifying a destination platform to the software application.
10. 4. The method of claim 3 wherein the machine language is compatible with the identified destination platform.
11. 5. The method of claim 3 wherein the destination platform comprises a processor architecture and an Operating System (OS).
12. 6. The method of claim 4 wherein the OS can be one of Microsoft Windows™ based OS, UNIX™ based OS, and Linux based OS, a real-time OS or a proprietary OS.
13. 7. The method of claim 1 wherein the model representation is a graphical representation.
14. 8. The method of claim 7 wherein the graphical representation conforms to the Unified Modeling Language (UML™) symbolic representation.
15. 9. The method of claim 1 wherein the model representation is a textual representation.

- 1 10. The method of claim 1 wherein the identified programming language can be one of:
2 Java™, C/C++, Ada, ALGOL, Assembly, COBOL, FORTRAN, Pascal, Perl, PL/I, Basic
3 and family (Visual Basic™, Quick Basic™), PHP, ASP, Delphi™, SQL, CGI, XML,
4 HTML, WAP or a proprietary programming language.
- 1 11. The method of claim 1 wherein the step of identifying a programming language for each
2 one of the plurality of software modules further comprises step of generating the
3 corresponding source code.
- 1 12. The method of claim 11 wherein the step of compiling each one of the plurality of
2 software modules into machine language using a software compiler further comprises
3 compiling the generated source code.
- 1 13. The method of claim 1 wherein a plurality of software compilers corresponding to the
2 identified programming language are used.
- 1 14. The method of claim 1 wherein the software compiler is a complete independent software
2 application.
- 1 15. The method of claim 1 wherein the software compiler is an incorporated software
2 application.
- 1 16. A tool for constructing a software application, the tool comprising:
 - 2 - an interpreting module for identifying a plurality of programming languages in
3 a source code listing; and
 - 4 - a calling module for compiling the source code listing into machine language.
- 1 17. The tool of claim 16 wherein the source code listing is contiguous or separated in multiple
2 parts.
- 1 18. The tool of claim 16 wherein the calling module uses a plurality of appropriate software
2 compilers for compiling each one of the plurality of programming languages.
- 1 19. The tool of claim 18 wherein at least one of the plurality of software compilers are
2 complete independent software applications.
- 1 20. The tool of claim 18 wherein at least one of the plurality of software compilers is
2 incorporated in the tool.